Flow Formulation-based Model for the Curriculum-based Course Timetabling Problem

Bagger, Niels-Christian Fink; Kristiansen, Simon; Sørensen, Matias; Stidsen, Thomas Jacob Riis

Published in:
MISTA 2015 Proceedings

Publication date:
2015

Document Version
Peer reviewed version

MISTA 2015 manuscript No.
(will be inserted by the editor)

Flow Formulation-based Model for the Curriculum-based Course Timetabling Problem

Niels-Christian Fink Bagger · Simon Kristiansen · Matias Sørensen · Thomas R. Stidsen

the date of receipt and acceptance should be inserted later

Abstract In this work we will present a new mixed integer programming formulation for the curriculum-based course timetabling problem. We show that the model contains an underlying network model by dividing the problem into two models and then connecting the two models back into one model using a maximum flow problem. This decreases the number of integer variables significantly and improves the performance compared to the basic formulation. It also shows competitiveness with other approaches based on mixed integer programming from the literature and improves the currently best known lower bound on one data instance in the benchmark data set from the second international timetabling competition.

1 Introduction

Each semester universities face the problem of generating high quality course timetables. A timetable determines when and where a course should take place. The problem of focus in this work is the Curriculum-based Course Timetabling (CCT) Problem from track 3 of the second international timetabling competition (ITC2007) as described by Gaspero et al (2007), in which weekly lectures for multiple courses have to be scheduled and assigned to rooms. A week is divided into days and each day is divided into time slots. A day and time slot combination is referred to as a period. The schedule and room assignment must fulfil some specific hard constraints; all lectures must be scheduled and in different periods, one

Niels-Christian Fink Bagger
Department of Management Engineering, Technical University of Denmark
E-mail: nbag@dtu.dk

Simon Kristiansen
Better Sports ApS

Matias Sørensen
MaCom A/S

Thomas R. Stidsen
Department of Management Engineering, Technical University of Denmark

teacher cannot give two lectures in the same period and a room cannot accommodate two lectures in the same period. Furthermore some courses are grouped into curricula and for each curriculum the courses within cannot be scheduled in the same periods.

Besides the hard constraints there are also soft constraints for which it is wanted to minimize the violation of these. For every lecture it is wanted to be able to accommodate a seat for each student attending. This is denoted as the *RoomCapacity* constraint and when a lecture is scheduled in a room the number of students above the capacity is the counted violation. Each course has a wish for the minimum number of days to spread the lectures across. This is denoted as the *MinimumWorkingDays* constraint and each day below this number in which lectures are not scheduled is counted as one violation. It is wanted to schedule lectures from the same curriculum in adjacent periods. Two periods are considered to be adjacent if they belong to the same day and are in consecutive time slots. If a lecture from a curriculum is scheduled in a period and no lecture from the same curriculum is scheduled in an adjacent period, the lecture is denoted as being *secluded*. Every time there is a secluded lecture this counts as one violation of the *CurriculumCompactness* constraint. Each course should not be assigned to too many different rooms during the week. This is denoted as the *RoomStability* constraint and every room except the first that the course is scheduled in is counted as one violation.

The objective is to find a solution which fulfils all the hard constraints and minimizes a weighted sum of the violations of the soft constraints. The problem will be solved using integer programming and the formulation of the model will be given in Section 2.

2 Mixed Integer Programming Formulation

The problem has been considered using mixed integer programming models before in the literature, see e.g. Burke et al (2010, 2012); Lach and Lübbecke (2012); Cacchiani et al (2013); Hao and Benlic (2011). For an great survey refer to Bettinelli et al (2015). A very common way to formulate the model is to use three-indexed binary variables. Here we will give the formulation similar to the three-indexed formulations from Burke et al (2012) and Lach and Lübbecke (2012). Let C be the set of courses, P be the set of periods and R be the set of rooms. Furthermore there are days D, curricula Q, lecturers L, the periods $P_d \subseteq P$ that belongs to day $d \in D$, the courses $C_q \subseteq C$ which are part of curriculum $q \in Q$ and the courses $C_l \subseteq C$ which are all being taught by lecturer $l \in L$. For each period $p \in P$ we will denote the adjacent periods as $p-1$ and $p+1$ for the periods belonging to the same day as p in the time slot right before and the time slot right after p respectively. When p corresponds to the first (last) time slot on the day, then the period $p-1$ $(p+1)$ is undefined and we will define any variable associated with it to always take the value zero.

Let L_c be the number of lectures to be scheduled for course $c \in C$, C_r be the capacity of room $r \in R$, S_c be the number of students attending course $c \in C$ and let $F_{c,p}$ be one if it is allowed to schedule a lecture from course $c \in C$ in period $p \in P$ and zero otherwise. Lastly M_c is the minimum number of days that it is preferred to schedule lectures for course $c \in C$ in.

Let $x_{c,p,r}$ be a binary variable deciding whether to schedule a lecture from course $c \in C$ in period $p \in P$ and room $r \in R$ or not. $t_{c,d}$ is a non-negative variable taking value 1 if course $c \in C$ has at least one lecture at day $d \in D$, and 0 otherwise. w_c is a non-negative variable denoting the number of days below the given minimum that course $c \in C$ has lectures. $z_{c,r}$ is a non-negative variable taking value 1 if course $c \in C$ is occupying room $r \in R$ at least once during the week, and 0 otherwise. κ_c is a non-negative variable counting the number of times that course $c \in C$ is changing room. $s_{q,p}$ is a non-negative variable taking value 1 if curriculum $q \in Q$ has a secluded lecture in period $p \in P$. Let W^{RC}, W^{CC}, W^{WD} and W^{RS} be the weights of the constraints *RoomCapacity*, *CurriculumCompactness*, *MinimumWorkingDays* and *RoomStability* respectively. The formulation is given in Model 1.

$$\min \quad W^{RC} \sum_{c\in C, p\in P, r\in R} (S_c - C_r)^+ \cdot x_{c,p,r} + W^{CC} \sum_{q\in Q, p\in P} s_{q,p} + W^{WD} \sum_{c\in C} w_c + W^{RS} \sum_{c\in C} \kappa_c \tag{1a}$$

$$\text{s. t.} \quad \sum_{p\in P, r\in R} x_{c,p,r} = L_c \qquad \forall c \in C \tag{1b}$$

$$\sum_{r\in R} x_{c,p,r} \leq F_{c,p} \qquad \forall c \in C, p \in P \tag{1c}$$

$$\sum_{c\in C} x_{c,p,r} \leq 1 \qquad \forall p \in P, r \in R \tag{1d}$$

$$\sum_{c\in C_q, r\in R} x_{c,p,r} \leq 1 \qquad \forall q \in Q, p \in P \tag{1e}$$

$$\sum_{c\in C_l, r\in R} x_{c,p,r} \leq 1 \qquad \forall l \in L, p \in P \tag{1f}$$

$$t_{c,d} - \sum_{p\in P_d, r\in R} x_{c,p,r} \leq 0 \qquad \forall c \in C, d \in D \tag{1g}$$

$$w_c + \sum_{d\in D} t_{c,d} \geq M_c \qquad \forall c \in C \tag{1h}$$

$$\sum_{p\in P} x_{c,p,r} - L_c \cdot z_{c,r} \leq 0 \qquad \forall c \in C, r \in R \tag{1i}$$

$$\sum_{r\in R} z_{c,r} - \kappa_c \leq 1 \qquad \forall c \in C \tag{1j}$$

$$\sum_{c\in C_q, r\in R} (x_{c,p,r} - x_{c,p-1,r} - x_{c,p+1,r}) \leq s_{q,p} \qquad \forall q \in Q, p \in P \tag{1k}$$

$$x_{c,p,r} \in \mathbb{B} \qquad \forall c \in C, p \in P, r \in R \tag{1l}$$

$$z_{c,r} \in \mathbb{B} \qquad \forall c \in C, r \in R \tag{1m}$$

$$0 \leq t_{c,d} \leq 1 \qquad \forall c \in C, d \in D \tag{1n}$$

$$w_c \geq 0 \qquad \forall c \in C \tag{1o}$$

$$\kappa_c \geq 0 \qquad \forall c \in C \tag{1p}$$

$$s_{q,p} \geq 0 \qquad \forall q \in Q, p \in P \tag{1q}$$

Model 1 A three-index formulation of the CCT problem.

The objective function (1a) consists of the weighted sum of the soft constraint violations and the weights are set according to Gaspero et al (2007):

$$W^{RC} = 1 \tag{1}$$
$$W^{CC} = 2 \tag{2}$$
$$W^{WD} = 5 \tag{3}$$
$$W^{RS} = 1 \tag{4}$$

The constraints (1b) ensures that all lectures of the courses are scheduled. Constraints (1c) ensures that each lecture of a course is scheduled in different periods and only in periods where the course is available. Constraints (1d) make sure that at most one lecture is scheduled in a room in any period. (1e) and (1f) ensures that courses from the same curriculum or taught by the same lecturer is not scheduled in the same periods. The constraints (1g) and (1h) computes which days that the course have been scheduled for lectures and by how much the minimum working days is violated. Constraints (1i) and (1j) calculates which rooms the courses puts into use and how many different rooms they are scheduled in. Lastly the constraints (1k) computes the periods where the curricula have secluded lectures.

2.1 Maximum Flow-based Formulation

The mixed integer programming formulation that we will present here is inspired by the formulation proposed by Lach and Lübbecke (2008, 2012) that consists of decomposing the model into two stages; stage I which is assigning time slots to the courses and stage II which is allocating rooms to the courses based on the assigned time slots from stage I. Instead of solving the two stages separately as Lach and Lübbecke (2008, 2012) we will combine the two stages into one model by using a flow network. This creates new models with a much lower number of integer variables compared to Model 1 at the cost of introducing three-indexed continuous variables. However due to the huge reduction in integer variables (and non-zeros in the constraint matrix) we expect these flow-based models to perform better than Model 1.

At first we will consider only assigning the courses to time slots, i.e. ignore the existence of rooms. This will only account for the *MinimumWorkingDays* and the *CurriculumCompactness* soft constraints. Let $x_{c,t}$ be a binary variable deciding whether to assign course $c \in C$ to time slot $t \in T$ or not. $t_{c,d}$, w_c and $s_{q,t}$ are defined in the same way as for Model 1. The formulation of assigning the courses to time slots is given in Model 2. The description of the objective and constraints follows that of Model 1.

The next step is to consider the room assignment part of the problem. Let $z_{c,r}$ be a binary variable taking value one if course $c \in C$ is allowed to be scheduled in room $r \in R$ and zero otherwise. Let κ_c be a non-negative variable counting the number of times that course $c \in C$ is changing room and let the integer variable $y_{c,r}$ identify the number of times that course $c \in C$ is assigned to room $r \in R$. The formulation is given in Model 3.

Constraints (3d) in Model 3 ensures that for some course $c \in C$ and some room $r \in R$, $z_{c,r}$ is set to one if $y_{c,r} > 0$. Constraints (3c) ensures that the total number

$$\min \quad W^{CC} \sum_{q \in Q, p \in P} s_{q,p} + W^{WD} \sum_{c \in C} w_c \tag{2a}$$

$$\text{s. t.} \quad \sum_{p \in P} x_{c,p} = L_c \qquad \forall c \in C \tag{2b}$$

$$x_{c,p} \leq F_{c,p} \qquad \forall c \in C, p \in P \tag{2c}$$

$$\sum_{c \in C_q} x_{c,p} \leq 1 \qquad \forall q \in Q, p \in P \tag{2d}$$

$$\sum_{c \in C_l} x_{c,p} \leq 1 \qquad \forall l \in L, p \in P \tag{2e}$$

$$\sum_{c \in C_q} (x_{c,p} - x_{c,p-1} - x_{c,p+1}) \leq s_{q,p} \qquad \forall q \in Q, p \in P \tag{2f}$$

$$t_{c,d} - \sum_{t \in T_d} x_{c,t} \leq 0 \qquad \forall c \in C, d \in D \tag{2g}$$

$$w_c + \sum_{d \in D} t_{c,d} \geq M_c \qquad \forall c \in C \tag{2h}$$

$$x_{c,p} \in \mathbb{B} \qquad \forall c \in C, p \in P \tag{2i}$$

$$s_{q,p} \geq 0 \qquad \forall q \in Q, p \in P \tag{2j}$$

$$0 \leq t_{c,d} \leq 1 \qquad \forall c \in C, d \in D \tag{2k}$$

$$w_c \geq 0 \qquad \forall c \in C \tag{2l}$$

Model 2 The formulation for assigning only the time slots.

$$\min \quad W^{RC} \sum_{c \in C, r \in R} (S_c - C_r)^+ \cdot y_{c,r} + W^{RS} \sum_{c \in C} p_c \tag{3a}$$

$$\text{s. t.} \quad \sum_{r \in R} z_{c,r} - p_c \leq 1 \qquad \forall c \in C \tag{3b}$$

$$\sum_{r \in R} y_{c,r} = L_c \qquad \forall c \in C \tag{3c}$$

$$y_{c,r} - L_c \cdot z_{c,r} \leq 0 \qquad \forall c \in C, r \in R \tag{3d}$$

$$y_{c,r} \in \mathbb{N} \qquad \forall c \in C, r \in R \tag{3e}$$

$$z_{c,r} \in \mathbb{B} \qquad \forall c \in C, r \in R \tag{3f}$$

$$p_c \geq 0 \qquad \forall c \in C \tag{3g}$$

Model 3 The formulation ignoring the time aspects and considering only the room stability and the room capacity violations.

of times that a course $c \in C$ is occupying some rooms is equal to the number of lectures to be taught.

If a solution $\overline{x}$ to Model 2 and a solution $\overline{y}$ to Model 3 is given then a new problem emerges; is the combined solution feasible, i.e. is there a feasible mapping from the assigned rooms in $\overline{y}$ to the assigned periods in $\overline{x}$ such that no room is occupied by two courses in the same period and no course is teaching two lectures in the same period. As a first attempt on the flow problem to use for the connection between Model 2 and Model 3 it is tempting to create a graph mapping the course-

room assignment $\overline{y}$ into room-period pairs. For each course $c \in C$ and each room $r \in R$ create a node (c, r) and for each room $r \in R$ and period $p \in P$ create a node (r, p). For each course $c \in C$, room $r \in R$ and period $p \in P$ create an arc from (c, r) to (r, p). Create a source node (u) and a sink node (v) and for each $c \in C$ and $r \in R$ create an arc from node (u) to node (c, r) and for every $r \in R$ and every $p \in P$ create an arc from node (r, p) to node (v).

The capacity on the arc $(r, p) \rightarrow (v)$ for some $r \in R$ and some $p \in P$ is one and always going to be unchanged. The remaining capacities are set based to some solution $(\overline{x}, \overline{y})$ for Model 2 and Model 3. For each course $c \in C$ and room $r \in R$ the capacity of the arc $(u) \rightarrow (c, r)$ is set to $\overline{y}_{c,r}$ and for each $c \in C$, $r \in R$ and $p \in P$ the capacity of the arc $(c, r) \rightarrow (r, t)$ is set to $\overline{x}_{c,p}$. An example of the graph is illustrated in Fig. 1.

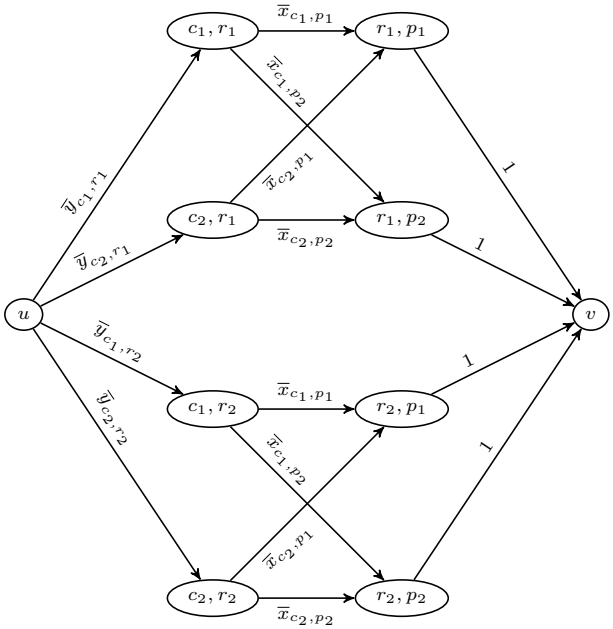

Fig. 1 Illustration of an attempt of the maximum flow graph of an instance with two courses, two rooms and two periods.

For each $c \in C$, $r \in R$ and $p \in P$ the amount of flow on the arc $(c, r) \rightarrow (r, p)$ in the graph in Fig. 1 corresponds to the number of times course c is assigned to room r in period p. Due to the capacities on the arcs at most one amount of flow can go through a node corresponding to a room and period pair (r, p), i.e. at most one course can be assigned to a room $r \in R$ in period $p \in P$. If the maximum flow in this graph is equal to the total sum of lectures of all courses then the solution $(\overline{x}, \overline{y})$ would be identified as being feasible. However, this is not always true. As an example consider an instance with three courses c_1 with one lecture, c_2 with two lectures and c_3 with one lecture, two periods p_1 and p_2, and two rooms r_1 and r_2. Consider a solution $(\overline{x}, \overline{y})$ assumed to be feasible for Model 2 and Model 3 where course c_1 has been assigned one lecture to room r_1 and assigned to period

p_1, course c_2 has been assigned one lecture to room r_1 and one lecture to room r_2 and has been assigned both period p_1 and p_2, course c_3 has been assigned to room r_2 and period p_1. The example is illustrated in Fig. 2 where only the arcs with positive capacities are illustrated.

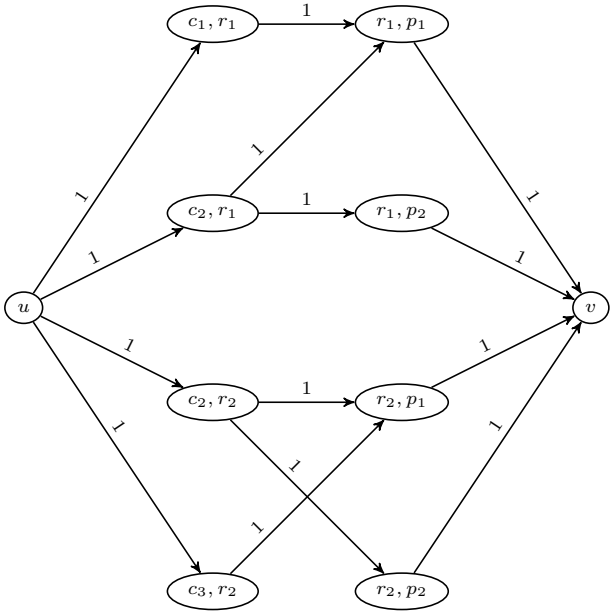

Fig. 2 Illustration of an example of the graph from Fig. 1 with three courses (c_1, c_2 and c_3), two periods (p_1 and p_2) and two rooms (r_1 and r_2). The capacities on the arcs illustrates the room assignments and the period assignments. This graph incorrectly deems the assignments feasible.

In Fig. 2 it can be seen that to get all lectures assigned the flow on the arcs from (u) must all equal the respective capacity. To get the flow out of node (c_1, r_1) it will have to be send to node (r_1, p_1) and to get the flow out of node (c_3, r_2) it will have to be send to node (r_2, p_1). This means that both of the rooms are occupied in period p_1. Since course c_2 has two lectures and there are only two periods then clearly the assignment is infeasible since the course cannot be assigned a room in period p_1. However it is possible to send all the flow through the graph. This is done by sending the flow that comes into node (c_2, r_1) further on to node (r_1, p_2) and sending the flow from node (c_2, r_2) to node (r_2, p_2). By this flow the course c_2 has been assigned two lectures in the same period in two different rooms which is an infeasible assignment for Model 1. However, since the value of the maximum flow is equal to the total number of lectures then this graph is incorrectly stating that the assignments are feasible. Therefore the graph needs to be extended to only allow one unit of flow for each course-period pair. For every room $r \in R$ and period $p \in P$ remove the arc $(r, p) \rightarrow (v)$ and split the node (r, p) into two nodes $(r, p)^1$ and $(r, p)^2$ and add an arc from $(r, p)^1$ to $(r, p)^2$ with a capacity of one. For every course $c \in C$ and period $p \in P$ create a node (c, p), add an arc to node (v) with a capacity of $\overline{x}_{c,p}$ and then for every room $r \in R$ add an arc from node $(r, p)^2$

to node (c, p) with capacity 1. The graph, denoted $\mathcal{G}_{\text{mf}}$, is illustrated in Fig. 3 where the nodes denoted $(r, p)^1$ are to the left in the graph and the nodes denoted $(r, p)^2$ are to the right.

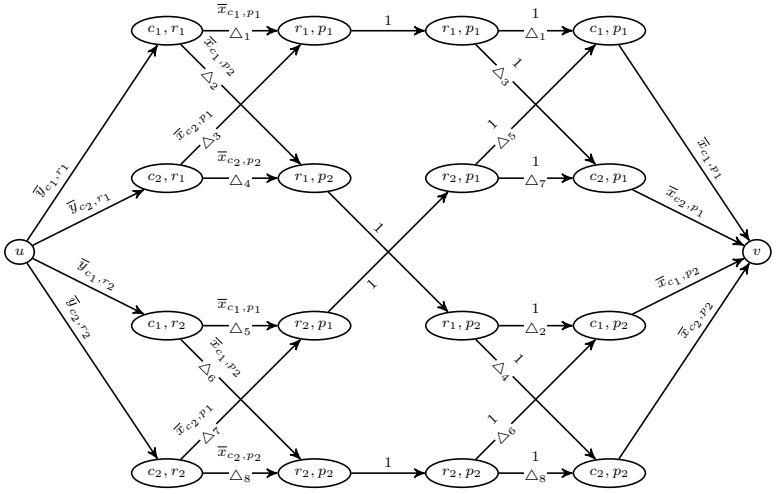

Fig. 3 Illustration of the maximum flow graph of an instance with two courses, two rooms and two periods.

Let the following non-negative variables be defined:

$f^u_{c,r}$: The amount of flow on the arc $(u) \to (c, r)$.
$f^1_{c,p,r}$: The amount of flow on the arc $(c, r) \to (r, p)^1$.
$f_{r,p}$: The amount of flow on the arc $(r, p)^1 \to (r, p)^2$.
$f^2_{c,p,r}$: The amount of flow on the arc $(r, p)^2 \to (c, p)$.
$f^v_{c,p}$: The amount of flow on the arc $(c, p) \to (v)$.

For a course $c \in C$, room $r \in R$ and period $p \in P$ the variable $f^1_{c,p,r}$ is indicating whether course c has a lecture scheduled in period p and room r, but so is the variable $f^2_{c,p,r}$. This means that, for the graph to be correct, if there exists an integer feasible flow f where the total amount of flow is equal to $\sum_{c \in C} L_c$ then there has to exist a flow f' with the same total amount of flow (it is possible that f' is the same flow as f) where $f^1_{c,p,r} = f^2_{c,p,r}$ for every triple (c, p, r). This is illustrated in the graph by marking the pair of arcs with the symbol Δ_i. If two arcs have the same Δ_i symbol then the flow on these two arcs must be equal. This is not taken care of in the standard formulation of the maximum flow problem, but this is not an issue by applying Proposition 1.

Proposition 1 *Let the total amount of flow (the value) of f be denoted $v(f)$ and let A be the set of feasible period-room assignments. Consider the (possibly fractional) maximum flow $f_{\max}$ in $\mathcal{G}_{mf}$ for a given period-room assignment pair $(\overline{x}, \overline{y})$. Then we have the following:*

$$v\left(f_{\max}\right) \geq \sum_{c \in C} L_c \iff (\overline{x}, \overline{y}) \in A$$

To prove Proposition 1 we will first show that $(\overline{x}, \overline{y}) \in A \implies v(f_{\max}) \geq \sum_{c \in C} L_c$. Next we will show that $v(f_{\max}) \geq \sum_{c \in C} L_c \implies (\overline{x}, \overline{y}) \in A$ by using Proposition 2.

Proposition 2 *Consider some period-room assignment pair $(\overline{x}, \overline{y})$ and let $F(\mathcal{G}_{mf})$ denote all feasible integer flows in $\mathcal{G}_{mf}$ given this assignment. If there exists a flow $f \in F(\mathcal{G}_{mf})$ where $v(f) \geq \sum_{c \in C} L_c$ then there exists a flow $f' \in F(\mathcal{G}_{mf})$ where $v(f') = v(f)$ and $f^1_{c,p,r} = f^2_{c,p,r}\ \forall c \in C, p \in P, r \in R$*

The proof of Proposition 2 is given in Appendix A.

Proof (Proof of $(\overline{x}, \overline{y}) \in A \implies v(f_{\max}) \geq \sum_{c \in C} L_c$ from Proposition 1) Assume that $(\overline{x}, \overline{y}) \in A$ and consider some feasible solution for this assignment. Let the variable $f_{c,p,r}$ take value one if course $c \in C$ is assigned to period $t \in T$ and room $r \in R$ in the considered solution. Since we are considering a feasible solution and it is based on the assignment $(\overline{x}, \overline{y})$ then the following conditions must be met:

$$\sum_{p \in P} f_{c,p,r} = \overline{y}_{c,r} \quad \forall c \in C, r \in R \tag{5}$$

$$\sum_{c \in C} f_{c,p,r} \leq 1 \quad \forall p \in P, r \in R \tag{6}$$

$$\sum_{r \in R} f_{c,p,r} = \overline{x}_{c,p} \quad \forall c \in C, p \in P \tag{7}$$

We will create a flow f' on the graph $\mathcal{G}_{\text{mf}}$ in the following way: Note that for each course $c \in C$, period $p \in P$ and room $r \in R$ there is a unique path $(u) \to (c,r) \to (r,p)^1 \to (r,p)^2 \to (c,p) \to (v)$ corresponding to the variable $f_{c,p,r}$ and if $f_{c,p,r} = 1$ then we will send one unit of flow on this path, otherwise not. Since we are only considering paths then the node balance constraints must all hold for this flow. Since $\sum_{p \in P} f_{c,p,r} = \overline{y}_{c,r}$ for some course $c \in C$ and room $r \in R$ then the total flow on the arc $(u) \to (c,r)$ is equal to $\overline{y}_{c,r}$ which is the capacity on that arc so the capacity cannot be exceeded. Since $\sum_{c \in C} f_{c,p,r} \leq 1$ for some room $r \in R$ and period $p \in P$ then the total amount of flow on the arc $(r,p)^1 \to (r,p)^2$ cannot exceed one which is the capacity on that arc so the flow is also feasible for this arc. This also means that the flow on the arc $(r,p)^2 \to (c,p)$ cannot exceed one for some course $c \in C$, period $p \in P$ and room $r \in R$ which is the capacity on this arc. Lastly since $\sum_{r \in R} f_{c,p,r} = \overline{x}_{c,p}$ for some $c \in C$ and $p \in P$ then the total flow going through the arc $(c,p) \to (v)$ must be equal to $\overline{x}_{c,p}$ which is the capacity on that arc. Furthermore this also means that the flow on the arc $(c,r) \to (r,p)^1$ can at most be $\overline{x}_{c,p}$ which is the capacity on that arc. This concludes that the flow we created must be a feasible flow for $\mathcal{G}_{\text{mf}}$ with respect to $(\overline{x}, \overline{y})$. Since $\sum_{r \in R} f_{c,p,r} = \overline{x}_{c,p}$ for every course $c \in C$ and period $p \in P$ and $\sum_{p \in P} \overline{x}_{c,p} = L_c$ then $\sum_{c \in C, p \in P, r \in R} f_{c,p,r} = \sum_c L_c$ and since each $f_{c,p,r}$ variable corresponds to a path then the total amount of flow $v(f')$ must be equal to $\sum_{c \in C} L_c$. Since f' is a feasible flow then the maximum flow f must have at least the same total amount of flow in $\mathcal{G}_{\text{mf}}$, so we have $v(f) \geq L_c$.

Proof (Proof of $v(f_{\max}) \geq \sum_{c \in C} L_c \implies (\overline{x}, \overline{y}) \in A$ from Proposition 1) The integrality requirements of the maximum flow problem can be removed from the

mathematical model since all capacities are integral (Ahuja et al, 1993, Theorem 6.5). By using (Ahuja et al, 1993, Theorem 6.5) then there must exist a maximum flow f with integer values and if $v(f) \geq \sum_{c \in C} L_c$ then Proposition 2 shows that there must exist an integer maximum flow where $f^1_{c,p,r} = f^2_{c,p,r}$ for every triple (c, p, r). This means that the $f^1_{c,p,r}$ variables and the $f^2_{c,p,r}$ variables describe a feasible assignment based on the solution pair $(\overline{x}, \overline{y})$ implying that $(\overline{x}, \overline{y}) \in A$.

The LP formulation of the maximum flow model as a feasibility problem, i.e. replacing the objective with a constraint that the value of the flow must be at least the number of lectures, is given in Model 4. We have substituted any occurrence of the flow variables $f^1_{c,p,r}$ and $f^2_{c,p,r}$ in Model 4, since we are only interested in a solution where the two variables are equal, with the non-negative variable $f_{c,p,r}$.

$$\sum_{c \in C, r \in R} f^u_{c,r} \geq \sum_{c \in C} L_c \tag{4a}$$

$$f^u_{c,r} - \sum_{p \in P} f_{c,p,r} = 0 \qquad \forall c \in C, r \in R \tag{4b}$$

$$\sum_{c \in C} f_{c,p,r} - f_{r,p} = 0 \qquad \forall p \in P, r \in R \tag{4c}$$

$$f_{r,p} - \sum_{c \in C} f_{c,p,r} = 0 \qquad \forall p \in P, r \in R \tag{4d}$$

$$\sum_{r \in R} f_{c,p,r} - f^v_{c,p} = 0 \qquad \forall c \in C, p \in P \tag{4e}$$

$$0 \leq f^u_{c,r} \leq \overline{y}_{c,r} \qquad \forall c \in C, r \in R \tag{4f}$$

$$0 \leq f_{c,p,r} \leq 1 \qquad \forall c \in C, p \in P, r \in R \tag{4g}$$

$$0 \leq f_{r,p} \leq 1 \qquad \forall p \in P, r \in R \tag{4h}$$

$$0 \leq f^v_{c,p} \leq \overline{x}_{c,p} \qquad \forall c \in C, p \in P \tag{4i}$$

Model 4 The feasibility flow problem.

Any solution to Model 4 can only fulfil constraint (4a) if the flow send out of the source node on each arc is equal to the capacity so the variable $f^u_{c,r}$ can be replaced with the value $\overline{y}_{c,r}$ in Model 4 which means that constraints (4a) and (4b) are replaced by:

$$\sum_{p \in P} f_{c,p,r} = \overline{y}_{c,r} \quad \forall c \in C, r \in R$$

Constraints (4c) and (4d) and the variable bounds (4h) can be replaced by the constraints:

$$\sum_{c \in C} f_{c,p,r} \leq 1 \quad \forall p \in P, r \in R$$

Finally the constraints (4e) and variable bounds (4g) and (4i) can be replaced by the constraints:

$$\sum_{r \in R} f_{c,p,r} \leq \overline{x}_{c,p} \quad \forall c \in C, p \in P$$

These latter mentioned substitutions together with Model 2 and Model 3 can then be combined into Model 5.

$$\begin{aligned}
\min \quad & W^{RC} \sum_{c,r} (S_c - C_r)^+ \cdot y_{c,r} + W^{CC} \sum_{q \in Q, p \in P} s_{q,p} \\
& + W^{WD} \sum_{c \in C} w_c + W^{RS} \sum_{c \in C} \kappa_c && \text{(5a)} \\
\text{s. t.} \quad & \text{(2b) — (2l)} && \text{(5b)} \\
& \text{(3b) — (3g)} && \text{(5c)} \\
& \sum_{p \in P} f_{c,p,r} = y_{c,r} \quad \forall c \in C, r \in R && \text{(5d)} \\
& \sum_{r \in R} f_{c,p,r} \leq x_{c,p} \quad \forall c \in C, p \in P && \text{(5e)} \\
& \sum_{c \in C} f_{c,p,r} \leq 1 \quad \forall p \in P, r \in R && \text{(5f)} \\
& f_{c,p,r} \geq 0 \quad \forall c \in C, p \in P, r \in R && \text{(5g)}
\end{aligned}$$

Model 5 The combined formulation connecting the period assignments and room assignments using the maximum flow model.

It is not guaranteed that the $f_{c,p,r}$ variables are integers in the solution obtained from Model 5. If the solution returned by the model contains fractional values for the $f_{c,p,r}$ variables then a polynomial algorithm to find an integer feasible solution can be applied. Such an algorithm is given in Algorithm 1 in Appendix A.

3 Computational Results

We have tested the model on the 21 data sets from the ITC2007 competition track 3 described in Gaspero et al (2007). Along the competition a benchmarking tool was provided. The benchmarking tool calculates the amount of time that the algorithms where allowed to run in the competition. This amount of time is usually referred to as one CPU time unit. We ran the tests in Windows 8.1 on an 3.07GHz Intel® Core™ i7 CPU with 12GB memory. Running the benchmarking tool returned 260 seconds as one CPU unit. All tests has been limited to a single thread.

As mentioned it may be needed to run some flow algorithms on the solutions returned by Model 5. The running times of these algorithms are just a matter of milliseconds even for the largest datasets and so we have neglected these algorithms from the time limits. Furthermore for all our tests the final solutions did not contain any fractional variables so the algorithms were never put to use. If $F_{c,p} = 0$ for some course $c \in C$ and period $p \in P$ then we do not add the variables $x_{c,p}$, $\{f_{c,p,r}\}_{r \in R}$ and $\{x_{c,p,r}\}_{r \in R}$ to the models. This makes the constraints (1c) and (2c) redundant since every course is taught by exactly one lecturer and constraints (1f) and (2e) ensures that each lecturer has at most one lecture scheduled in any period. Furthermore we replace the constraints (1e), (1f), (2d) and (2e) by clique

inequalities. This is done by creating a graph where each node corresponds to a course. An edge is connecting two courses if they are in the same curriculum or taught be the same lecturer. We then enumerate all the maximal cliques by running the BronKerbosch algorithm Bron and Kerbosch (1973). Let Γ be the set of cliques and let C_γ be the set of courses in the clique $\gamma \in \Gamma$. Then for each clique $\gamma \in \Gamma$ and period $p \in P$ we add the following constraints to both the basic model and the maximum flow-based formulation:

$$\sum_{c \in C_\gamma, r \in R} x_{c,p,r} \leq 1 \quad \forall \gamma \in \Gamma, p \in P$$
$$\sum_{c \in C_\gamma} x_{c,p} \leq 1 \quad \forall \gamma \in \Gamma, p \in P$$

Enumerating all the maximal cliques takes less than a second even for the largest data instances we have tested so we have neglected these enumerations from the time limits when solving the models.

In Table 1 the statistics of the basic model and the maximum flow-based formulation can be seen. For each of the 21 data sets the number of continuous variables, integer variables, constraints and non-zeros is reported.

Table 1: The statistics of the different models of the 21 test instances from ITC2007 track 3; the basic formulation (Basic) and the maximum flow-based formulation (MF). For each data instance and formulation the number of continuous variables (Cont.), the number of integer variables (Int.), the number of rows in the model (Rows) and the number of non-zeros (Non-Zeros) is reported. The number in parenthesis denotes how many of the integer variables that are binary (Bin.).

		Cont.	Int. (Bin.)	Rows	Non-Zeros
comp01	Basic	630	5580 (5580)	2670	61620
	MF	5712	1207 (1027)	2794	29544
comp02	Basic	2324	34112 (34112)	9593	672958
	MF	26916	4161 (2849)	10032	134643
comp03	Basic	2204	29952 (29952)	8628	593908
	MF	24892	3722 (2570)	8973	123222
comp04	Basic	1978	36972 (36972)	8104	528387
	MF	30400	4423 (3001)	8833	141108
comp05	Basic	5436	17982 (17982)	14256	868140
	MF	15993	2145 (1659)	11927	98302
comp06	Basic	2506	50544 (50544)	12333	940504
	MF	39730	5956 (4012)	13223	194715
comp07	Basic	2842	68120 (68120)	15368	1220537
	MF	55002	7848 (5228)	16825	264695
comp08	Basic	2127	40248 (40248)	8186	511003
	MF	32223	4768 (3220)	8897	146461

Continued on next page

Table 1 – Continued from previous page

		Cont.	Int. (Bin.)	Rows	Non-Zeros
comp09	Basic	2407	35568 (35568)	8851	604293
	MF	29317	4231 (2863)	9413	137921
comp10	Basic	2480	53820 (53820)	13665	1036975
	MF	41738	6321 (4251)	14593	206834
comp11	Basic	795	6900 (6900)	3855	88395
	MF	7075	1556 (1406)	3910	40313
comp12	Basic	6104	35816 (35816)	22712	1666356
	MF	25904	3736 (2768)	19368	165972
comp13	Basic	2224	40508 (40508)	8114	553425
	MF	32282	4698 (3140)	8937	147523
comp14	Basic	2095	37570 (30570)	9875	699435
	MF	29958	4529 (3084)	10514	148453
comp15	Basic	2204	29952 (29950)	8628	593908
	MF	24892	3722 (2570)	8973	123222
comp16	Basic	2531	56160 (56160)	12599	1013091
	MF	46171	6502 (4342)	13783	220565
comp17	Basic	2443	43758 (43758)	11050	778699
	MF	35202	5293 (3610)	11836	171848
comp18	Basic	2248	15651 (15651)	7290	304912
	MF	12130	1944 (1521)	6147	62416
comp19	Basic	2168	30784 (30784)	7926	490106
	MF	24168	3743 (2559)	8306	115114
comp20	Basic	2797	59774 (59774)	14067	1103445
	MF	47143	6932 (4633)	15180	228938
comp21	Basic	2608	43992 (43992)	11919	952542
	MF	36574	5271 (3579)	12777	182969
Change	Average	x12.5	x0.1 (x0.1)	x1.0	x0.2
	Min	x2.9	x0.1 (x0.1)	x0.8	x0.1
	Max	x19.4	x0.2 (x0.2)	x1.1	x0.5

It can be seen in Table 1 that the maximum flow-based formulation increases the number of continuous variables on average more that 12 times. However the number of integer variables and non-zeroes in the model is on average a tenth and a fifth respectively compared to the basic model which is why we expect the maximum flow-based formulation to perform better. The model has been solved using the .NET framework provided by Gurobi Optimization (2015) version 6.0.0. The bounds obtained by the maximum flow-based formulation is compared on the first 14 data sets with the following four approaches from the literature:

LL12 An approach based on solving the problem in two stages proposed by Lach and Lübbecke (2012); first assigning the courses into periods and then assigning the first stage assignments into rooms.

BMPR10 A approach based on considering a subset of soft constraints proposed by Burke et al (2010).

HB11 An approach based on partitioning the courses into subsets proposed by Hao and Benlic (2011).

CCRT13 An approach based on splitting the objective into parts proposed by Cacchiani et al (2013).

In Table 2 the lower bounds for the latter mentioned four approaches and the maximum flow-based formulation is reported when running the approaches for one CPU unit (1 T), ten CPU units (10 T) and forty CPU units (40 T). It can be seen that the proposed formulation is able to compete with most of the approaches, except for the proposed method by Cacchiani et al (2013) which seems to perform better on most instances. However the maximum-flow based formulation appears to generate a much better bound on two of the instances; comp05 and comp12. Referring back to Table 1 these are the two only of the first fourteen instances where the formulation actually reduces instead of increasing the number of rows in the model. Furthermore consider Table 5. In this table the number of courses and the number of unavailable time slots are illustrated for each instance. Here it can be seen that the number of unavailable time slots per course is much higher for the two before mentioned instances than for the other of the first fourteen data sets. This can explain why the number of rows is reduced since we did not include the variables of the unavailable periods and so many rows where not added as they were empty.

Since Lach and Lübbecke (2012) and Burke et al (2010) obtain both lower and upper bound these are also compared with the bounds obtained by the maximum flow-based formulation. The results are given in Table 3. Here it can be seen that Burke et al (2010) obtains better lower bounds in most cases for one CPU unit, however for longer running times the maximum flow formulation generates better lower bound on more instances than the other two. As for the upper bounds Lach and Lübbecke (2012) yields better result in more cases than our proposed approach for the short (1 T) and middle (10 T) running time whereas for the long (40 T) running time they yield better upper bounds on an equal amount fo instances making it hard to claim one approach as outperforming the other.

In Table 4 the results of both the basic formulation in Model 1 and the maximum flow based formulation is given. Here it can be seen that the maximum flow formulation clearly outperforms the basic formulation and a new lower bound compared to the best known bound is obtained in one of the instances. This makes the model very interesting as some of the other approaches from the literature based in the basic formulation might also benefit from this reformulation.

	LL12			BMPR10			HB11			CCRT13			MF		
Instance	1 T	10 T	40 T	1 T	10 T	40 T	1 T	10 T	40 T	1 T	10 T	40 T	1 T	10 T	40 T
comp01	4	4	4	0	4	**5**	4	4	4	**5**	**5**	**5**	**5**	**5**	**5**
comp02	0	8	11	0	0	1	**10**	12	12	0	**16**	**16**	0	0	10
comp03	0	0	25	25	33	33	**26**	34	36	24	**52**	**52**	**26**	35	36
comp04	22	28	28	**35**	**35**	**35**	**35**	**35**	**35**	**35**	**35**	**35**	23	**35**	**35**
comp05	92	25	108	**119**	111	114	19	69	80	6	6	166	**119**	**171**	**179**
comp06	7	10	10	**13**	**15**	**16**	12	12	**16**	0	11	11	**13**	13	**16**
comp07	0	2	**6**	**6**	**6**	**6**	5	**6**	**6**	0	**6**	**6**	0	**6**	**6**
comp08	30	34	**37**	**37**	**37**	**37**	**37**	**37**	**37**	**37**	**37**	**37**	27	**37**	**37**
comp09	37	41	46	68	65	66	39	67	67	**92**	**92**	**92**	45	71	76
comp10	2	**4**	**4**	3	**4**	**4**	**4**	**4**	**4**	0	2	2	3	**4**	**4**
comp11	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
comp12	29	32	53	**101**	95	95	43	78	84	0	0	100	85	**115**	**138**
comp13	33	39	41	52	52	54	46	53	55	**57**	**57**	**57**	38	54	56
comp14	40	41	46	**41**	42	42	**41**	43	43	32	**48**	**48**	**41**	42	46
Best	**1**	**2**	**4**	**8**	**6**	**7**	**7**	**5**	**6**	**6**	**10**	**10**	**6**	**8**	**9**
	0	**0**	**0**	**2**	**1**	**0**	**2**	**0**	**0**	**2**	**5**	**5**	**0**	**2**	**2**

Table 2 Comparison of the lower bounds obtained for the different model formulations; Lach and Lübbecke (2012) (LL12), Burke et al (2010) (BMPR10), Hao and Benlic (2011) (HB11), Cacchiani et al (2013) (CCRT13) and the maximum flow based formulation (MF). For each formulation the lower bound is given for one CPU time unit (1 T), ten CPU time units (10 T) and forty CPU time units (40 T). The numbers reported in bold font are the values where the specific models obtained a value which is at least as good as the other formulations. The numbers underlined are the values where the specific models obtained a value which is the better that for the other formulations.

	LL12						BMPR10						MF					
	1 T		10 T		40 T		1 T		10 T		40 T		1 T		10 T		40 T	
Instance	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
comp01	4	12	4	12	4	12	0	168	4	10	**5**	9	**5**	**5**	**5**	**5**	**5**	**5**
comp02	**0**	239	**8**	93	**11**	**46**	**0**	**114**	0	101	1	63	**0**	253	0	**74**	10	54
comp03	0	194	0	**86**	25	**66**	25	**158**	33	144	33	123	**26**	228	**35**	115	**36**	84
comp04	22	**44**	28	41	28	38	**35**	153	**35**	**36**	**35**	36	23	123	**35**	38	**35**	**35**
comp05	92	965	25	**468**	108	**368**	**119**	1447	111	649	114	629	**119**	**515**	**171**	505	**179**	377
comp06	7	395	10	**79**	10	51	**13**	**277**	**15**	317	**16**	**46**	**13**	897	13	298	**16**	71
comp07	0	**525**	2	**28**	**6**	**25**	**6**	-	**6**	857	**6**	45	0	1095	**6**	215	**6**	58
comp08	30	**78**	34	48	**37**	44	**37**	173	**37**	53	**37**	41	27	195	**37**	**44**	**37**	**40**
comp09	37	115	41	**106**	46	**99**	**68**	**112**	65	115	66	105	45	213	**71**	127	**76**	**99**
comp10	2	235	**4**	**44**	**4**	**16**	**3**	**70**	**4**	49	**4**	23	**3**	994	**4**	311	**4**	44
comp11	**0**	7	**0**	7	**0**	7	**0**	288	**0**	12	**0**	12	**0**	**0**	**0**	**0**	**0**	**0**
comp12	29	**1122**	32	657	53	548	**101**	-	95	889	95	785	85	1844	**115**	**507**	**138**	**485**
comp13	33	**98**	39	**67**	41	66	**52**	556	52	92	54	67	38	461	**54**	102	**56**	**65**
comp14	40	**113**	41	**54**	**46**	**53**	**41**	123	**42**	72	42	55	**41**	180	**42**	84	**46**	58
Best	**2**	**6**	**3**	**8**	**6**	**7**	**12**	**5**	**7**	**1**	**7**	**1**	**8**	**3**	**12**	**5**	**13**	**7**
	0	**6**	**1**	**8**	**1**	**6**	**6**	**5**	**1**	**1**	**0**	**1**	**2**	**3**	**6**	**5**	**5**	**6**

Table 3 Comparison of the bounds obtained by Lach and Lübbecke (2012) (LL12), Burke et al (2010) (BMPR10) and the maximum flow based formulation (MF). For each approach the lower bound is given for one CPU time unit (1 T), ten CPU time units (10 T) and forty CPU time units (40 T). The numbers reported in bold font are the values where the approach obtained a value which is at least as good as the other approaches. The numbers underlined are the values where the specific approaches obtained a value which is better than the other approaches.

			Basic						MF					
	Best Known		1 T		10 T		40 T		1 T		10 T		40 T	
Instance	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
comp01	5	5	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
comp02	16	24	**0**	308	**2**	139	5	71	**0**	**253**	0	**74**	**10**	**54**
comp03	52	64	25	1415	28	181	28	109	**26**	**228**	**35**	**115**	**36**	**84**
comp04	35	35	22	182	**35**	59	**35**	**35**	**23**	**123**	**35**	**38**	**35**	**35**
comp05	211	284	117	537	141	**484**	149	387	**119**	**515**	**171**	505	**179**	**377**
comp06	27	27	12	1403	12	**135**	14	124	**13**	**897**	**13**	298	**16**	**71**
comp07	6	6	**0**	**354**	**6**	315	**6**	119	**0**	1095	**6**	**215**	**6**	**58**
comp08	37	37	20	**177**	**37**	65	**37**	61	**27**	195	**37**	**44**	**37**	**40**
comp09	96	96	37	272	65	167	68	159	**45**	**213**	**71**	**127**	**76**	**99**
comp10	4	4	0	**256**	**4**	**94**	**4**	56	**3**	994	**4**	311	**4**	**44**
comp11	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
comp12	100	298	59	**1363**	69	717	106	581	**85**	1844	**115**	**507**	**138**	**485**
comp13	59	59	28	**280**	50	130	53	117	**38**	461	**54**	**102**	**56**	**65**
comp14	51	51	39	305	**42**	112	42	96	**41**	**180**	**42**	**84**	**46**	**58**
comp15	52	64	25	1415	28	181	28	109	**26**	**228**	**35**	**115**	**36**	**84**
comp16	18	18	**8**	**329**	8	103	11	101	7	400	**12**	**74**	**13**	**58**
comp17	56	56	24	**335**	30	268	39	155	**33**	450	**42**	**122**	**43**	**105**
comp18	61	61	14	168	22	165	26	103	**20**	**145**	**26**	**88**	**29**	**83**
comp19	57	57	30	**205**	49	143	52	138	**36**	210	**53**	**62**	**57**	**57**
comp20	4	4	**0**	**1169**	**0**	**120**	**0**	**103**	**0**	1215	**0**	972	**0**	**103**
comp21	74	74	15	847	31	258	49	186	**32**	**527**	**54**	**142**	**57**	**122**
Best			**6**	**11**	**9**	**6**	**7**	**4**	**20**	**12**	**20**	**17**	**21**	**21**
			2	**2**	**6**	**2**	**7**	**3**	**2**	**2**	**7**	**2**	**8**	**4**

Table 4 Comparison of the bounds obtained for the basic model (Basic) and the maximum flow-based formulation (MF). For each formulation the bounds are given for one CPU time unit (1 T), ten CPU time units (10 T) and forty CPU time units (40 T). The numbers reported in bold font are the values where the specific models obtained a value which is at least as good as the other formulations. The numbers underlined are the values where the specific models obtained a value which is as least as good as the best known bounds reported by Scheduling and Timetabling Research Group at the University of Udine (2015).

Instance	$\|C\|$	$\|U\|$	$\|U\|/\|C\|$
comp01	30	53	1.8
comp02	82	513	6.3
comp03	72	382	5.3
comp04	79	396	5.0
comp05	54	771	14.3
comp06	108	632	5.9
comp07	131	667	5.1
comp08	86	478	5.6
comp09	76	405	5.3
comp10	115	694	6.0
comp11	30	94	3.1
comp12	88	1368	15.5
comp13	82	468	5.7
comp14	85	486	5.7

Table 5 Illustrating the statistics of the data sets regarding the unavailable periods. For each instance the number of courses ($|C|$), the total number of unavailable periods ($|U|$) and the average number of unavailable periods per course ($|U|/|C|$) is reported.

4 Conclusion

A mixed integer programming model for the curriculum-based course timetabling problem has been proposed with an underlying flow network. It has been shown that the formulation decreases the number of integer variables significantly and provides better results than a traditional three-index formulation. It is also competitive with most of the other mixed integer programming based approaches from the literature and improves one currently best known lower bound on the benchmarking instances from the second international timetabling competition. Some of the approaches from the literature are based on the original three-indexed model and it is believed that these approaches can also benefit from the proposed model.

Acknowledgements The authors would like to thank professor Stephan Røpke, Department of Management Engineering, Technical University of Denmark, and professor Carsten Thomassen, Department of Applied Mathematics and Computer Science, Technical University of Denmark for fruitful discussions on the proof in Appendix A.

A Proof of Proposition 2

For the proof of Proposition 2 we will be considering the graph $\mathcal{G}_{\text{mf}}$ as described in Section 2.1 and a given solution pair $(\overline{x}, \overline{y})$ to Model 2 and Model 3 where $\sum_{p \in P} \overline{x}_{c,p} = \sum_{r \in R} \overline{y}_{c,r} = L_c \, \forall c \in C$ due to constraints (2b) and (3c). Furthermore we will assume that $L_c \leq |P| \; \forall c \in C$. This is a fair assumption to make as the problem is otherwise infeasible. Before the proposition is proved it will be restated here for the sake of completeness.

Proposition 3 (Restatement of Proposition2)
Consider some period-room assignment pair $(\overline{x}, \overline{y})$ and let $F\left(\mathcal{G}_{mf}\right)$ denote all feasible integer flows in $\mathcal{G}_{mf}$ given this assignment. If there exists a flow $f \in F\left(\mathcal{G}_{mf}\right)$ where $v(f) \geq \sum_{c \in C} L_c$ then there exists a flow $f' \in F\left(\mathcal{G}_{mf}\right)$ where $v(f') = v(f)$ and $f^1_{c,p,r} = f^2_{c,p,r} \, \forall c \in C, p \in P, r \in R$

Algorithm 1: EqualPairMaxFlow

Input: The graph $\mathcal{G}_{\text{mf}}$
Output: An integer maximum flow f where $f^1_{c,t,r} = f^2_{c,t,r}$ $\forall c \in C$, $t \in T$, $r \in R$ if $v(f) \geq \sum_{c \in C} L_c$, otherwise nil
Initialize f as the maximum flow in $\mathcal{G}_{\text{mf}}$
if $v(f) < \sum_{c \in C} L_c$ **then**
  return nil
`// Iterate over all` (c, p, r)`-triples to repair any violations`
foreach $c \in C$ **do**
  foreach $p \in P$ **do**
    foreach $r \in R$ **do**
      `/* Change the flow` f `by setting` $f^2_{c,p,r}$ `to the same value as` $f^1_{c,p,r}$ `*/`
      $f^2_{c,p,r} \leftarrow f^1_{c,p,r}$
return f

To prove Proposition 2 we will show that Algorithm 1 is correct.

Algorithm 1 starts off by finding a maximum flow f which has integer values. This can be done by some polynomial algorithm, e.g. the Labeling algorithm (Ahuja et al, 1993, proof of Theorem 6.5). If $v(f) < \sum_c L_c$ then the algorithm returns **nil** to indicate that the assignment $(\overline{x}, \overline{y})$ is infeasible which has been proved in Section 2.1 to be the case. If $v(f) \geq \sum_c L_c$ then the algorithm iterates over every triple (c, p, r) and then set the value of the variable $f^2_{c,p,r}$ to the same value as the variable $f^1_{c,p,r}$. When the algorithm is done then clearly the flow still maintains integer values and $f^1_{c,p,r} = f^2_{c,p,r}$ for every triple (c, p, r). What needs to be shown to prove that Algorithm 1 is correct is that the value of the flow is unchanged, the node balancing constraints are not violated and the capacities are not exceeded, i.e. that the the flow after the change remains a feasible flow.

Assuming that Algorithm 1 is correct we can prove Proposition 2.

Proof (Proof of Proposition 2) As Algorithm 1 is correct then Proposition 2 must be true since if $v(f) < \sum_{c \in C} L_c$ the algorithm returns that the assignment $(\overline{x}, \overline{y})$ is infeasible and if $v(f) \geq \sum_{c \in C} L_c$ the algorithm will return an integer maximum flow f where $f^1_{c,p,r} = f^2_{c,p,r}$ $\forall c \in C$, $p \in P$, $r \in R$.

To prove that Algorithm 1 is correct we will first show that when considering an integer feasible flow f where $v(f) \geq \sum_{c \in C} L_c$, if there is a violation then it is possible to redirect the flow such that the total number of violations is decreased by at least two as stated in Proposition 4. This means that if we have k violations for the flow f then applying this redirection technique at most $k/2$ times will remove all such violations where k must be less than or equal to $|C| \cdot |P| \cdot |R|$.

Proposition 4 *Consider a flow $f \in F\left(\mathcal{G}_{mf}\right)$ where $v(f) \geq \sum_{c \in C} L_c$. If there exists violations of some course-period-room triples (c, p, r), i.e. that $f^1_{c,p,r} \neq f^2_{c,p,r}$, then it is possible to redirect the flow to another flow $f' \in F\left(\mathcal{G}_{mf}\right)$ where $v(f') = v(f)$ such that the total number of violations is decreased by at least two.*

Proof (Proof of Proposition 4) Consider a flow $f \in F\left(\mathcal{G}_{\text{mf}}\right)$ where $v(f) \geq \sum_{c \in C} L_c$. Assume that there exists violations of some course-period-room triples (c, p, r), i.e. that $f^1_{c,p,r} \neq f^2_{c,p,r}$. Since we know that $\sum_{p \in P} \overline{x}_{c,p} = L_c$ for every course $c \in C$ then for every period $p \in P$ where $\overline{x}_{c,p} = 1$ there must be at least one unit of flow on the arc $(c, p) \rightarrow (v)$ otherwise all the flow from the source cannot get to the sink. Let there be a course-period-room triple (c_1, p_1, r_1) such that $f^1_{c_1,p_1,r_1} \neq f^2_{c_1,p_1,r_1}$. Since the capacity on the arc $(c_1, r_1) \rightarrow (r_1, p_1)^1$ is $\overline{x}_{c_1,p_1}$ which is a binary value and the capacity on the arc $(r_1, p_1)^2 \rightarrow (c_1, p_1)$ is one then two cases can occur:

Case 1 $f^1_{c_1,p_1,r_1} = 0 \wedge f^2_{c_1,p_1,r_1} = 1$

Case 2 $f^1_{c_1,p_1,r_1} = 1 \wedge f^2_{c_1,p_1,r_1} = 0$

Consider first Case 1. This case is illustrated in Fig. 4.

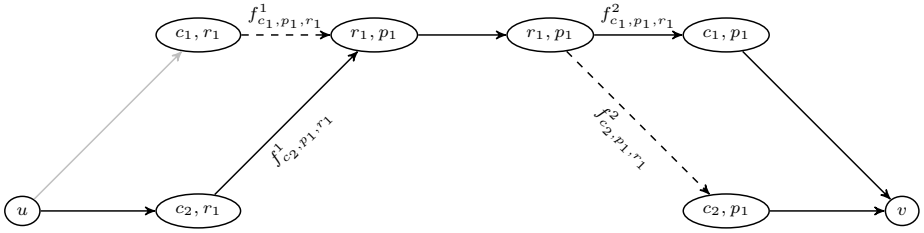

Fig. 4 Illustration of Case 1. The dashed arcs means that there is no flow. The lightly gray arcs correspond to where it is unknown whether there is any flow and the solid black arcs are where there must be at least one unit of flow.

Since $f^2_{c_1,p_1,r_1} = 1$ this means that $f_{r_1,p_1} = 1$ since this is the only way flow can enter the node $(r_1, p_1)^2$ meaning that node $(r_1, p_1)^1$ must be sending out one unit of flow. Since $(r_1, p_1)^1$ is sending out one unit of flow then it must mean that $f^1_{c_2,p_1,r_1} = 1$ for some course $c_2 \in C$. Furthermore since the capacity on the arc $(r_1, p_1)^1 \to (r_1, p_1)^2$ is one and this is the only arc entering $(r_1, p_1)^2$ then it cannot send any units of flow to node (c_2, p_1). This means that Case 1 must contain a triple (c_2, p_1, r_1) for which Case 2 applies. So we can prove the claim for both cases by only considering Case 2.

Consider now Case 2. Since $f^1_{c_1,p_1,r_1} = 1$ then there must be a unit of flow on the arc from $(r_1, p_1)^2$ to (c_2, p_1) for some course $c_2 \in C$. This is illustrated in Fig. 5.

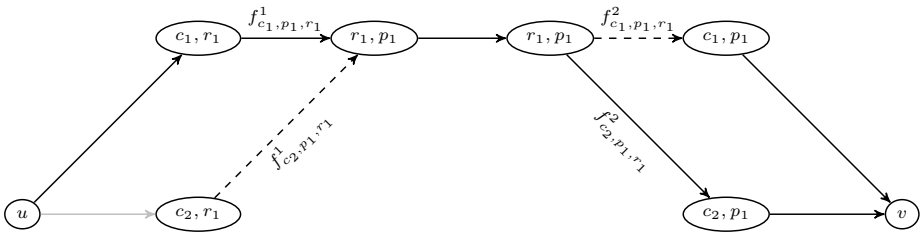

Fig. 5 Illustration of Case 2. The interpretation of the arcs corresponds to Fig. 4.

Due to the construction of the graph the capacity on the arc $(c_1, p_1) \to (v)$ is equal to the capacity on the arc $(c_1, r_1) \to (r_1, p_1)^1$ and must therefore be one since $f^1_{c_1,p_1,r_1} = 1$. This means that we could redirect the flow on the subpath $(r_1, p_1)^2 \to (c_2, p_1) \to (v)$ to the subpath $(r_1, p_1)^2 \to (c_1, p_1) \to (v)$ if there is no flow on the arc $(c_1, p_1) \to (v)$ and maintain an integer feasible flow where the total amount of flow is unchanged. However as latter mentioned the flow on the arc $(c_1, p_1) \to (v)$ must be one, i.e. $f^v_{c_1,p_1} = 1$, since $\overline{x}_{c_1,p_1} = 1$ which means that node (c_1, p_1) must receive one unit of flow from node $(r_2, p_1)^2$ for some room $r_2 \in R$. This case is illustrated in Fig. 6.

Consider the four nodes $(r_1, p_1)^2$, (c_1, p_1), $(r_2, p_1)^2$ and (c_2, p_1) in Fig. 6. As mentioned earlier the capacity on the arc $(r_1, p_1)^2 \to (c_1, p_1)$ must be one. The capacity on the arc $(r_2, p_1)^2 \to (c_2, p_1)$ is set to $\overline{x}_{c_2,p_1}$ which is also the capacity on the arc $(r_1, p_1)^2 \to (c_2, p_1)$ which must be one since there is one unit of flow on the arc $(r_1, p_1)^2 \to (c_2, p_1)$. So swapping the flow on the arc $(r_1, p_1)^2 \to (c_2, p_1)$ with the arc $(r_1, p_1)^2 \to (c_1, p_1)$ and swapping the flow on the arcs $(r_2, p_1)^2 \to (c_1, p_1)$ and $(r_2, p_1)^2 \to (c_2, p_1)$ will maintain an integer feasible

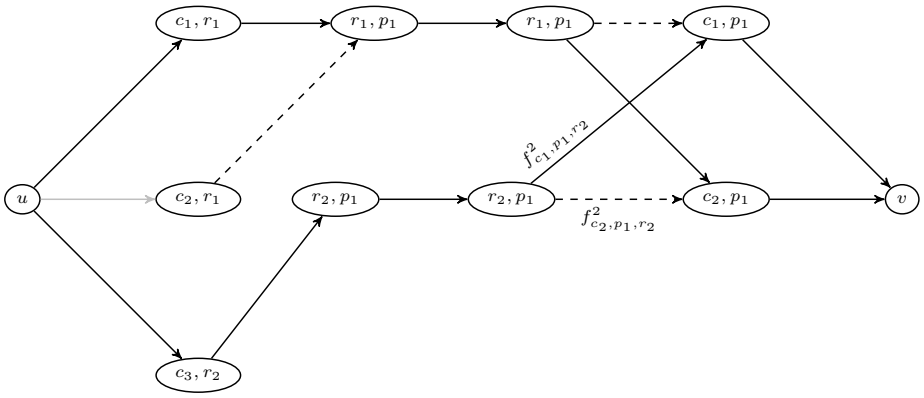

Fig. 6 Illustration of Case 2. The interpretation of the arcs corresponds to Fig. 4.

flow with an unchanged amount of flow where the Case 2 violation is removed from the triple (c_1, p_1, r_1) and the Case 1 violation is removed from the triple (c_2, p_1, r_1). This swap does not introduce new violations when $c_1 \neq c_3$ and so we are done. However if $c_1 = c_3$ then one violation is introduced for the triple (c_1, p_1, r_2) and one for the triple (c_2, p_1, r_2) meaning that making the swaps does not change the number of violations. So we need to show that when $c_1 = c_3$ it is possible to find another place in the graph to make the swap and repair at least two violations.

Since we know that there is flow from the source to the nodes (c_1, r_1) and (c_1, r_2) then we know that $\sum_{r \in R} \overline{y}_{c_1,r} \geq 2$ meaning that $\sum_{p \in P} \overline{x}_{c_1,p} \geq 2$, i.e. that c_1 is teaching at least two lectures. This means that there must exist another period $p_2 \in P : p_1 \neq p_2$ where $\overline{x}_{c_1,p_2} = 1$. As $\overline{x}_{c_1,p_2} = 1$ implies that there is at least one unit of flow on the arc $(c_1, t_2) \rightarrow (v)$ then there must be one unit of flow on the path $(u) \rightarrow (c_4, r_3) \rightarrow (r_3, p_2)^1 \rightarrow (r_3, p_2)^2 \rightarrow (c_1, p_2) \rightarrow (v)$ for some $c_4 \in C$ and $r_3 \in R$ as illustrated in Fig. 7.

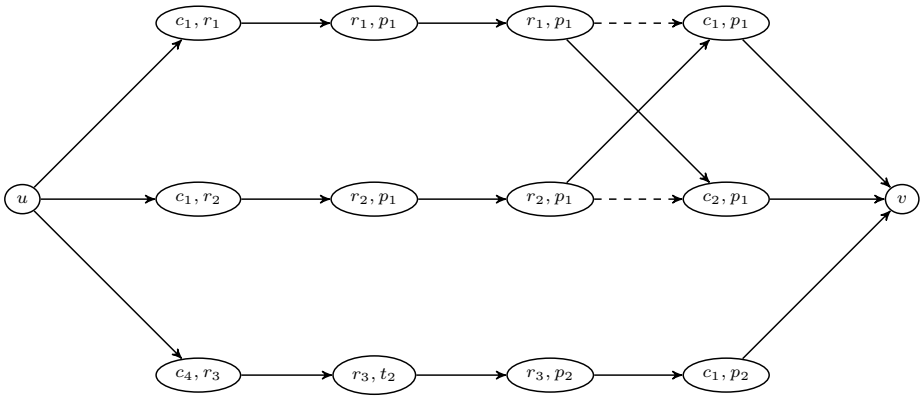

Fig. 7 Illustration of Case 2 where $c_1 = c_3$. The interpretation of the arcs corresponds to Fig. 4.

Suppose that $c_1 = c_4$. Then $\sum_{r \in R} y_{c_1,r} \geq 3$ and there must be some other period that c_1 is assigned to and we can consider that period as p_2 instead. So we must be able to find a period $p_2 \in T$ and a course $c_4 \in C$ such that $p_1 \neq p_2$ and $c_1 \neq c_4$ where there is flow on

the path $(u) \to (c_4, r_3) \to (r_3, p_2)^1 \to (r_3, p_2)^2 \to (c_1, p_2) \to (v)$ for some $r_3 \in R$. This means that for the triple (c_1, p_2, r_3) there is a Case 1 violation and for the triple (c_4, p_2, r_3) there is a Case 2 violation. This is exactly the same cases as for the triples (c_2, p_1, r_1) and (c_1, p_1, r_1) and so there must exist a course $c_5 \in C$ and a room $r_4 \in R$ where there is one unit of flow on the path $(u) \to (c_5, r_4) \to (r_4, p_2)^1 \to (r_4, p_2)^2 \to (c_4, p_2) \to (v)$. This means that we have two cases; either $c_4 \neq c_5$ and we can swap the flow on the arcs $(r_3, p_2)^2 \to (c_1, p_2)$ and $(r_3, p_2)^2 \to (c_4, p_2)$ and swap the flow on the arcs $(r_4, p_2)^2 \to (c_4, p_2)$ and $(r_4, p_2)^2 \to (c_1, p_2)$ or $c_4 = c_5$ and we can find a path $(u) \to (c_6, r_5) \to (r_5, p_3)^1 \to (r_5, p_3)^2 \to (c_4, p_3) \to (v)$ where there is one unit of flow and where $c_6 \neq c_4$ and $p_3 \neq p_2$ in the same way as we found c_4 and p_2. This is illustrated in Fig. 8.

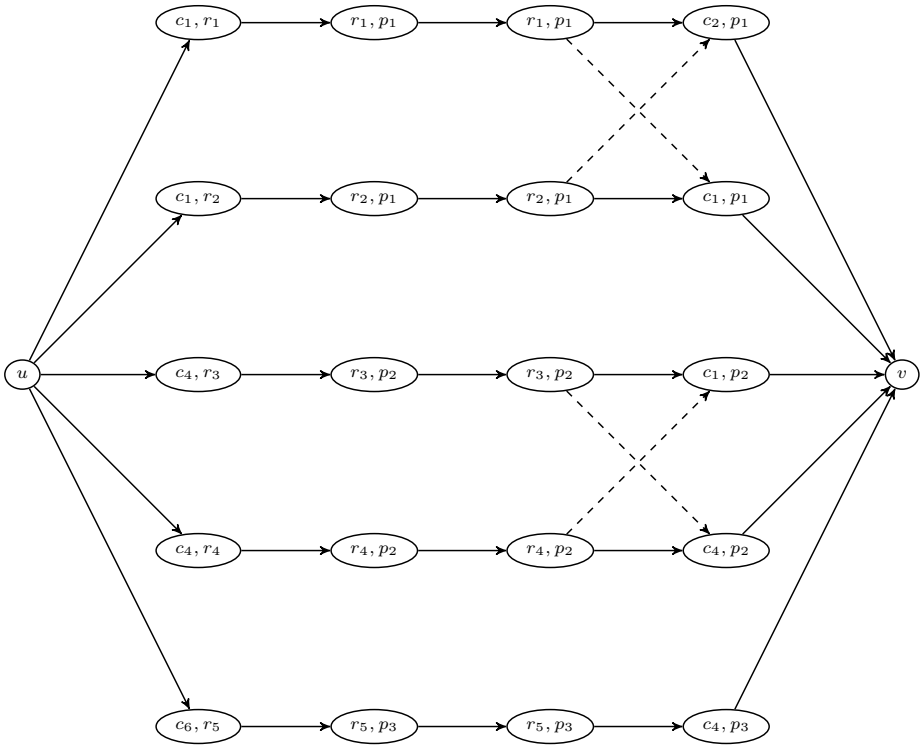

Fig. 8 Illustration of Case 2 after a couple of iterations. The interpretation of the arcs corresponds to Fig. 4.

It may be the case that $c_6 = c_1$. However this indicates that the course c_1 is teaching at least one more lecture than previously thought when we found the period p_2, so we can backtrack to the point where we found p_2 and then find another period instead of p_2 which we have not yet considered for c_1. This means that whenever we are considering a violating pair we must be able to either redirect the flow for that pair or find a new violating pair which involves a new course which we either have not yet considered before or it is a course we have been considering before and then we can backtrack to this course and consider a new period which we have not considered before for that course. Clearly all the operations can be made in polynomial asymptotic time but it needs to be shown that the total number of backtracking operations is finitely bounded to ensure that the algorithm is finite. So if the total number of backtrack operations is finite then eventually the algorithm will end up with some pair where the flow can be redirected and decrease the number of violations.

Let $T(m,n)$ be the total number of backtrack operations that our algorithm performs where $m = |P|$ and $n = |C|$. The number of times that we backtrack to the first course in our algorithm can at most be the number of lectures taught by the course since we consider a new period not considered for the course before whenever we backtrack. Since the number of lectures is linearly bounded by m then we can at most backtrack $O(m)$ times to the first course. Every time we backtrack to the first course we have been backtracking $T(m,n-1)$ times to the remaining courses meaning that we have the following recursive relation:

$$T(m,n) = O(m) \cdot T(m,n-1)$$

Consider when $n = 2$. We can backtrack to the first course $O(m)$ times but we can never backtrack to the second course since there are no other courses to backtrack from and so we have the base case:

$$T(m,2) = O(m)$$

We will now show that the recursion leads to a finite number of total backtracking operations by making a guess of the asymptotic bound:

$$T(m,n) = O\left(m^{n-1}\right)$$

It is easy to see that it holds for the base case so we can assume that it holds for $T(m,n-1)$ and then we have:

$$\begin{aligned} T(m,n-1) &= O\left(m^{n-2}\right) \\ T(m,n) &= O(m) \cdot O\left(m^{n-2}\right) = O\left(m^{n-1}\right) \end{aligned}$$

It has now been shown by induction that the algorithm is making a finite number of backtracking operations which concludes the proof of Proposition 4.

Before proving the correctness of Algorithm 1 it should be noted that since $\sum_{r \in R} \overline{y}_{c,r} = L_c$ $\forall c \in C$ then the total capacity on the outgoing arcs of the source is $\sum_{c \in C} L_c$. This means that for the maximum flow f it must always hold that $v(f) \leq \sum_{c \in C} L_c$. Furthermore since all capacities in the graph $\mathcal{G}_{\text{mf}}$ are integers then there must be a maximum flow taking integer values (Ahuja et al, 1993, Theorem 6.5).

Proof (Proof that Algorithm 1 is correct) The proof of Proposition 4 implies that if we have a feasible integer flow $f \in \mathcal{G}_{\text{mf}}$ where $v(f) \geq \sum_{c \in C} L_c$ and if for some triple (c,p,r) we have that $f^1_{c,p,r} \neq f^2_{c,p,r}$ then there must exist some courses $c_1 \in C$, $c_2 \in C$, $c_3 \in C$, some rooms $r_1 \in R$, $r_2 \in R$ and a period $p_1 \in P$ where $c_1 \neq c_3$ (it is possible that $c_2 = c_3$) and the following holds; $f^1_{c_1,p_1,r_1} = 1$, $f^1_{c_1,p_1,r_2} = 0$, $f^1_{c_2,p_1,r_1} = 0$, $f^1_{c_3,p_1,r_2} = 1$, $f^2_{c_1,p_1,r_1} = 0$, $f^2_{c_1,p_1,r_2} = 1$, $f^2_{c_2,p_1,r_1} = 1$ and $f^2_{c_2,p_1,r_2} = 0$. Swapping the values of $f^2_{c_1,p_1,r_1}$ and $f^2_{c_1,p_1,r_2}$ and swapping the values of $f^2_{c_2,p_1,r_1}$ and $f^2_{c_2,p_1,r_2}$ will remain an integer feasible flow with the same total amount of flow and a decrease in the number of violations by at least two. So a simple algorithm is to search for these courses, these rooms and this period where the swap can be done, do the swap and then iterate. This can be implemented to run in polynomial asymptotic time instead of the exponential asymptotic time given in the proof of Proposition 4. However, since the swaps are only done on the $f^2_{c,p,r}$ variables and never on the $f^1_{c,p,r}$ variables then this means that the values of the $f^1_{c,p,r}$ variables must be feasible values for the $f^2_{c,p,r}$ and so a much simpler algorithm can be constructed by the following steps:

- Step 1 Find an integer maximum flow f. This can be done by some polynomial maximum flow algorithm, e.g. the Labeling Algorithm (Ahuja et al, 1993, proof of Theorem 6.5, section 6.5)
- Step 2 If $v(f) < \sum_{c \in C} L_c$ then return **nil**, i.e. that it is infeasible, which is correct as it has been proved that the assignment pair $(\overline{x}, \overline{y})$ cannot be feasible in this case, otherwise go to Step 3.
- Step 3 Iterate over all triples (c,p,r) and set the value of the variable $f^2_{c,p,r}$ equal to the value of the $f^1_{c,p,r}$ variable and return this new flow.

Note that these steps is exactly the description of Algorithm 1 and so the algorithm must be correct.

References

Ahuja RK, Magnanti TL, Orlin JB (1993) Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., Upper Saddle River, NJ, USA

Bettinelli A, Cacchiani V, Roberti R, Toth P (2015) An overview of curriculum-based course timetabling. TOP pp 1–37

Bron C, Kerbosch J (1973) Algorithm 457: Finding all cliques of an undirected graph. Commun ACM 16(9):575–577, DOI 10.1145/362342.362367, URL http://doi.acm.org/10.1145/362342.362367

Burke E, Marecek J, Parkes A, Rudová H (2010) Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research 37(3):582–597

Burke EK, Marec J, Parkes AJ, Rudova H (2012) A branch-and-cut procedure for the udine course timetabling problem. Annals of Operations Research 194(1):71–87

Cacchiani V, Caprara A, Roberti R, Toth P (2013) A new lower bound for curriculum-based course timetabling. Computers & Operations Research 40(10):2466 – 2477

Gaspero LD, Schaerf A, McCollum B (2007) The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3). Tech. rep., School of Electronics, Electrical Engineering and Computer Science, Queens University SARC Building, Belfast, United Kingdom

Gurobi Optimization I (2015) Gurobi optimizer reference manual. URL http://www.gurobi.com

Hao JK, Benlic U (2011) Lower bounds for the itc-2007 curriculum-based course timetabling problem. European Journal of Operational Research 212(3):464 – 472

Lach G, Lübbecke M (2008) Optimal university course timetables and the partial transversal polytope. In: McGeoch C (ed) Experimental Algorithms, Lecture Notes in Computer Science, vol 5038, Springer Berlin / Heidelberg, pp 235–248

Lach G, Lübbecke M (2012) Curriculum based course timetabling: new solutions to udine benchmark instances. Annals of Operations Research 194:255–272

Scheduling, Timetabling Research Group at the University of Udine I (2015) Curriculum-based course timetabling. http://tabu.diegm.uniud.it/ctt/index.php